A BRIEF INTRODUCTION TO THE LEVEL SET METHODS

Richard Tsai
Department of Mathematics and PACM
Princeton University

- What is it for?
- What is it?
- How do I implement it?
- What are the applications?

The problem

Want a way to deal with curves in R^2 and surfaces in R^3.

The problem

Want a way to deal with curves in R^2 and surfaces in R^3.

Can be for interface dynamics, geometrically based motions, etc.

The problem

Want a way to deal with curves in R^2 and surfaces in R^3.

Can be for interface dynamics, geometrically based motions, etc.

Want:

- ✔ Good marriage with geometry
- ✔ Painless topological changes
- ✔ Efficient algorithms and good theory

Overview

- Level Set Methods consist of
  - Data representation: closed curves, surfaces, and sets, quantities on surfaces
  - Dynamics: moving curves, surfaces, and sets, changing quantities defined on surfaces
  - Numerical methods: finite difference methods

Overview

- Level Set Methods consist of
  - Data representation: closed curves, surfaces, and sets, quantities on surfaces
  - Dynamics: moving curves, surfaces, and sets, changing quantities defined on surfaces
  - Numerical methods: finite difference methods

- What can level set do for you?
  - Fix a surface, tracking quantities around or on it
  - Tracking quantities around or on moving surfaces

Overview

- Level Set Methods consist of
  - Data representation: closed curves, surfaces, and sets, quantities on surfaces
  - Dynamics: moving curves, surfaces, and sets, changing quantities defined on surfaces
  - Numerical methods: finite difference methods
- What can level set do for you?
  - Fix a surface, tracking quantities around or on it
  - Tracking quantities around or on moving surfaces
- What can't conventional level set methods do, YET?

What else is there?

- Phase field method
- Segment projection method (Engquist, Tornberg)
- DSE (Dynamic surface extension) (Steinhoff)
- Front tracking
- VOF (Volume of fluid)
- Recent work: Particle level set method (Enright-Fedkiw)

Where to find references and recent progress

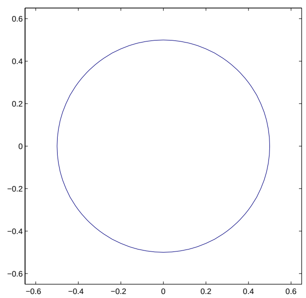
- The original paper: Osher-Sethian [1988]
- Book: Osher-Fedkiw, Springer 2002
- UCLA's CAM Report website: `http://www.math.ucla.edu/applied/cam/`
- Stanford CS group: `http://www.cs.stanford.edu/~fedkiw`
- UC Berkeley's math website

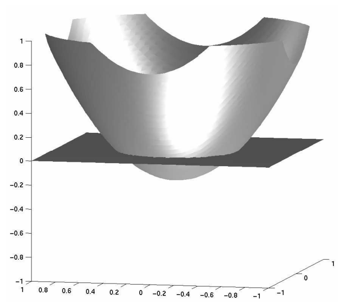
Level Set for data representation

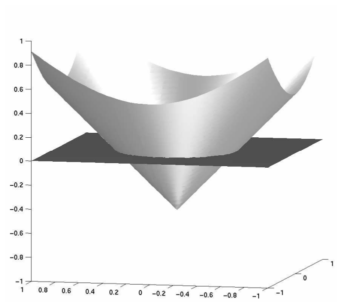
- Closed curves in R^2 and surfaces in R^3, and in general, codimension 1 objects, Γ, and regions enclosed Ω.

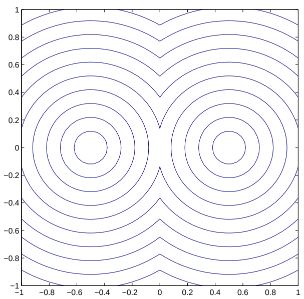
- Implicit: Γ is defined as the kernel of a Lipschitz continuous function ϕ; i.e.

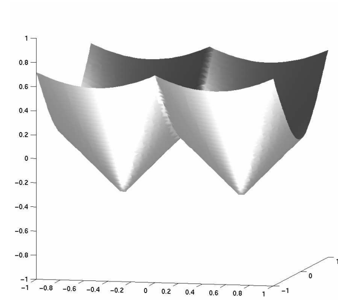
$$\Gamma = \{(x,y) \in R^2 : \phi(x,y) = 0\}$$

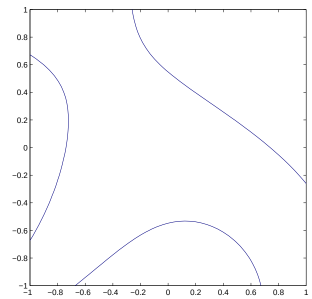


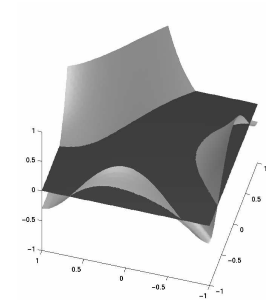


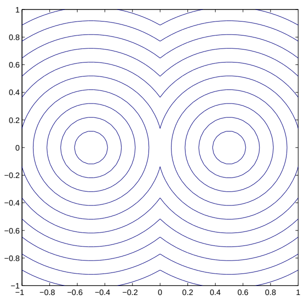


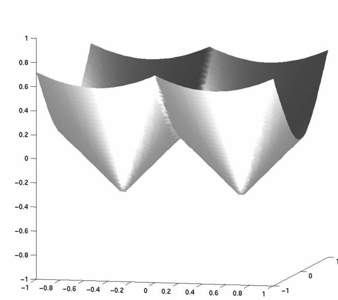

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1 −0.8 −0.6 −0.4 −0.2 0 0.2 0.4 0.6 0.8 1

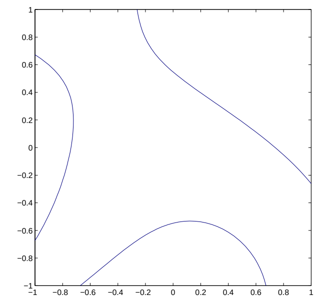


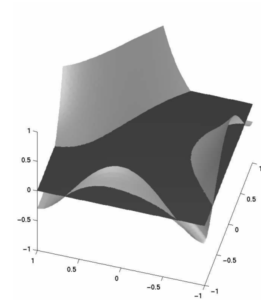

1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1 −0.8 −0.6 −0.4 −0.2 0 0.2 0.4 0.6 0.8 1











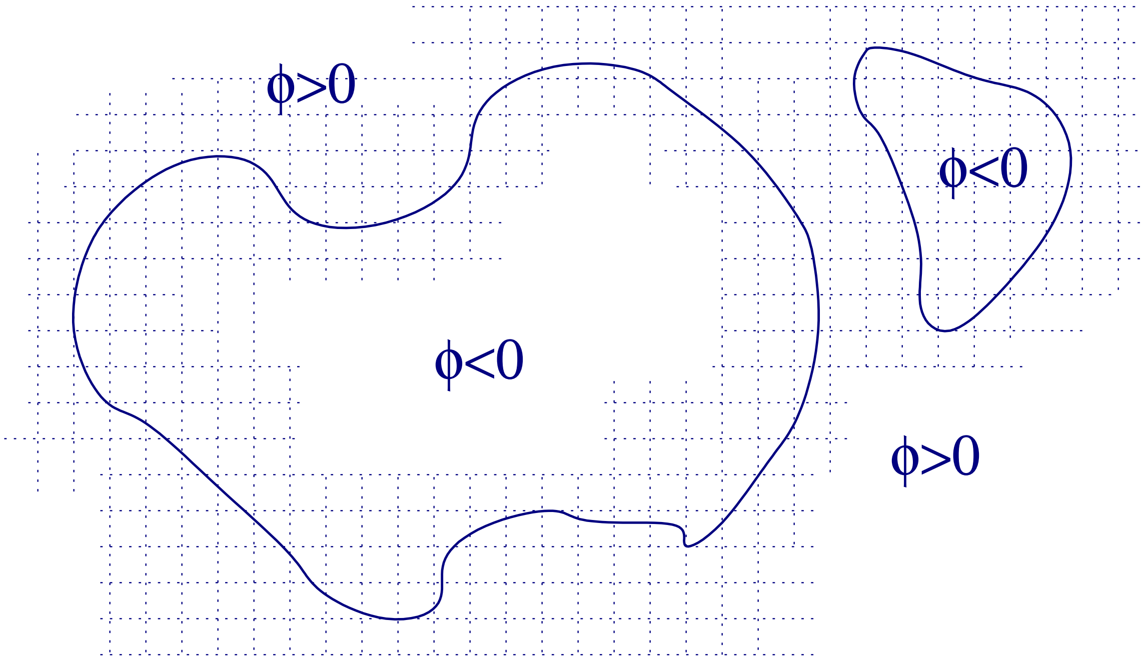
- Objects with higher codimensions (LT Cheng et al.)

- More complicated curves and regions. E.g. open curves (OCKHT, Smereka), multiple phases (Zhao, Vese-Chan)

Extraction of geometrical informations

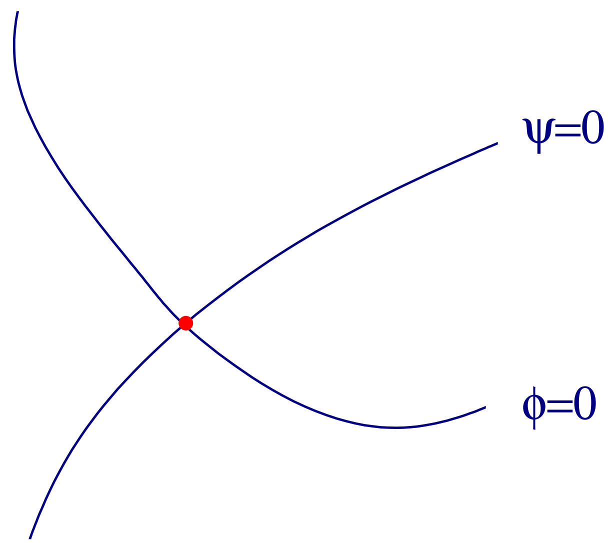
- Normals: $\pm\nabla\phi/|\nabla\phi|$
- Mean curvature: $\nabla \cdot n = \nabla \cdot (\nabla\phi/|\nabla\phi|)$
- Volume enclosed:
$$\int_\Omega H(-\phi(x))dx = \int_{\chi_{\{\phi<0\}}} 1dx$$
- Surface Integral:
$$\int_\Omega \delta(\phi)|\nabla\phi|dx$$

In real life...

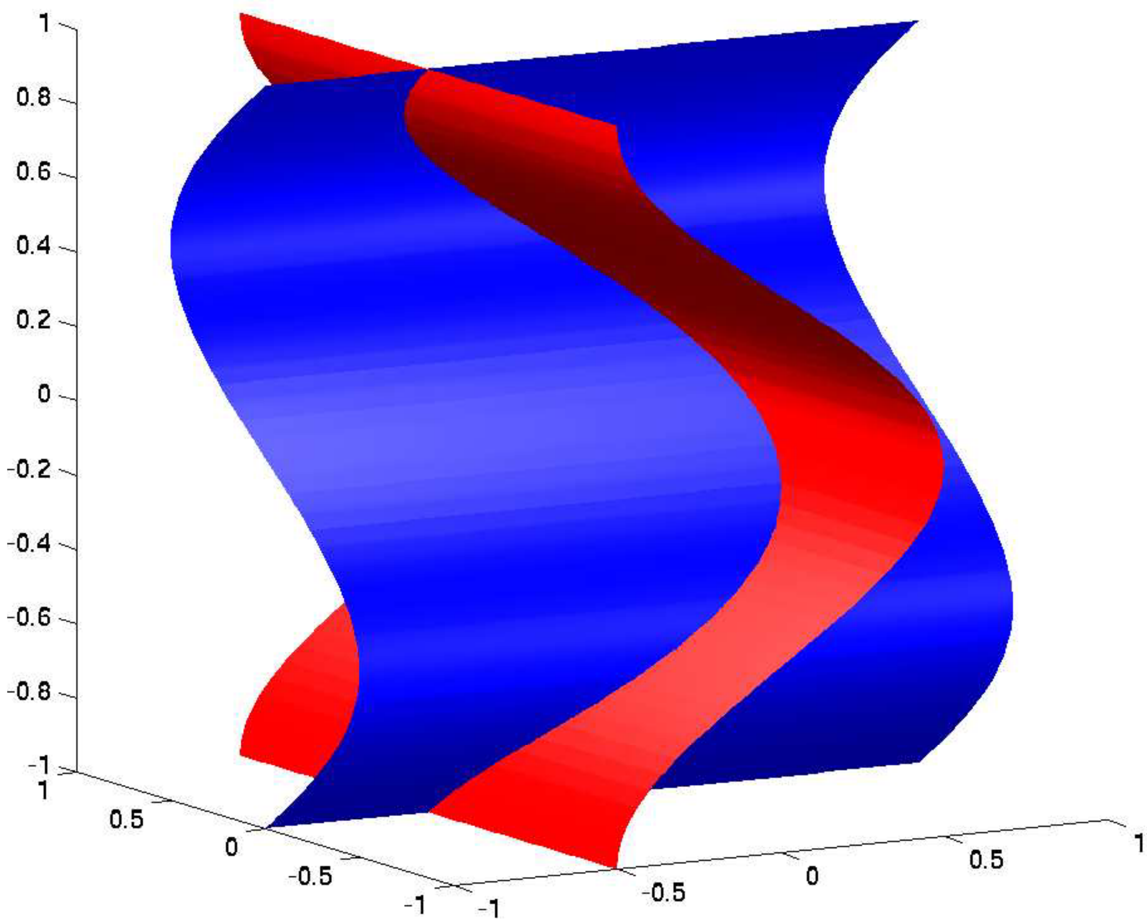


Geometrical quantities approximated by finite differences and numerical quadratures on a locally uniform grid.

Higher codimension objects

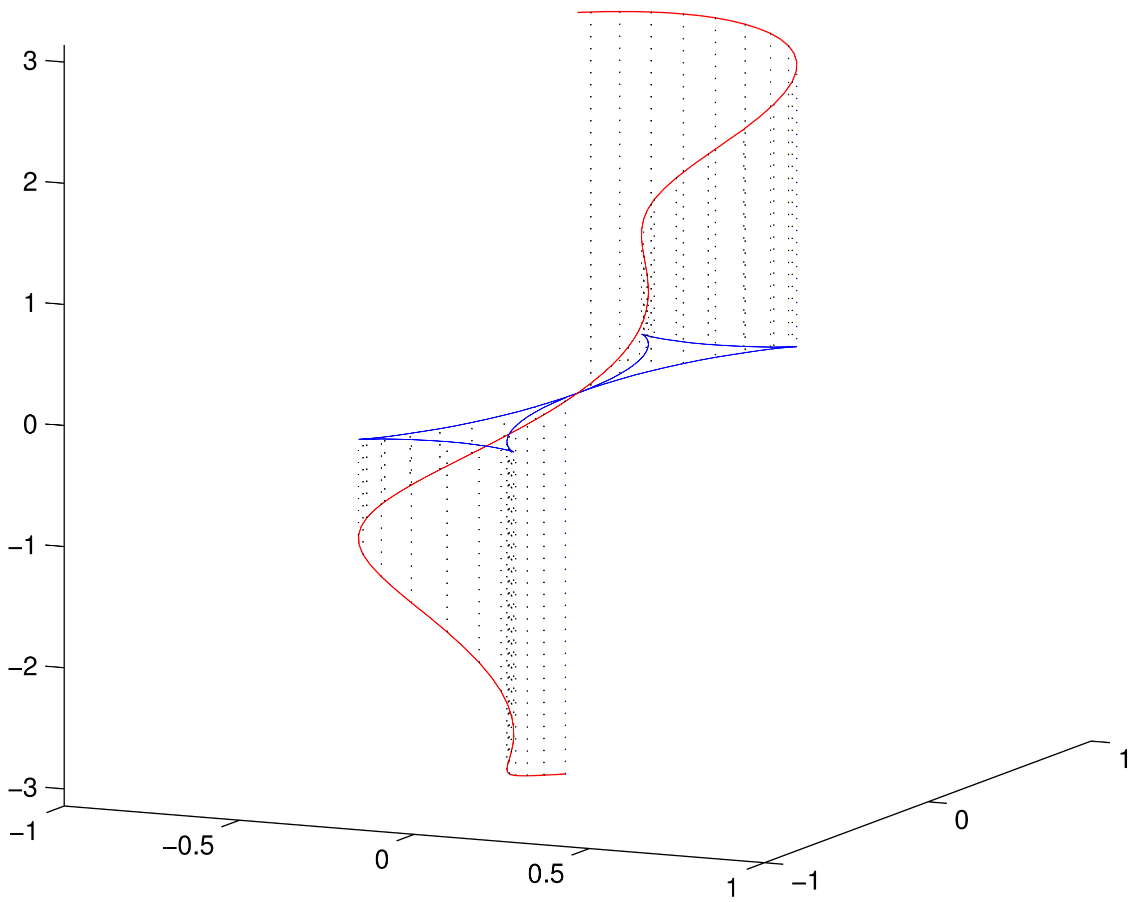


Curves in R^3



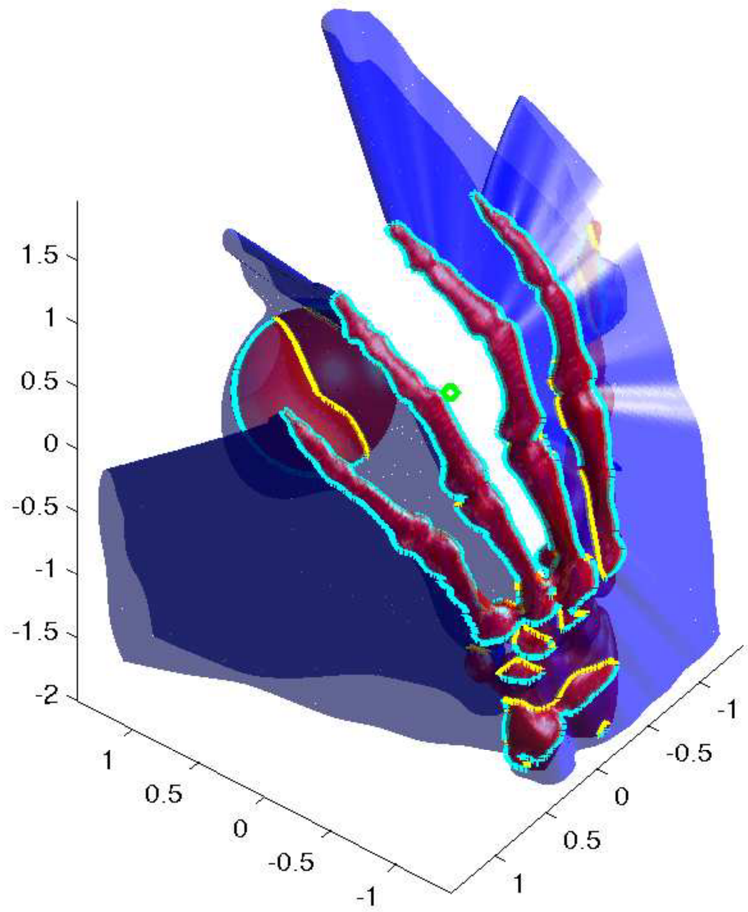
Path integral: $\int_\Omega f(x)\delta(\phi)\delta(\psi)|\nabla\phi \times \nabla\psi| dx$

Open or non-simple curves in R^2



Ref: Osher-Cheng-Kang-Shim-Tsai, J. of Comput. Phys. 2002.

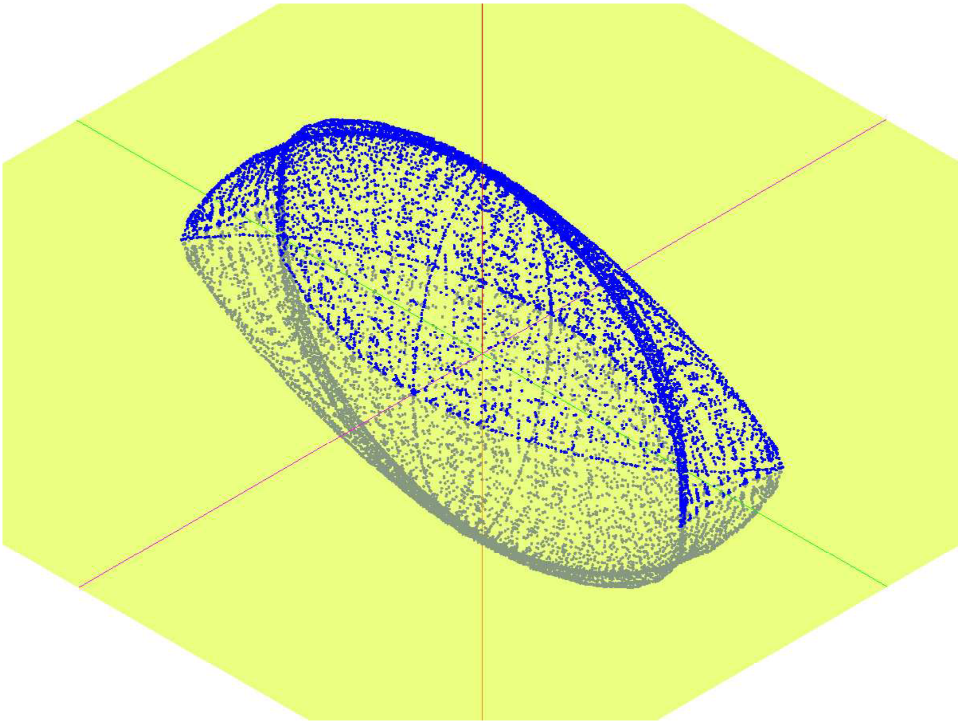
More complicated example



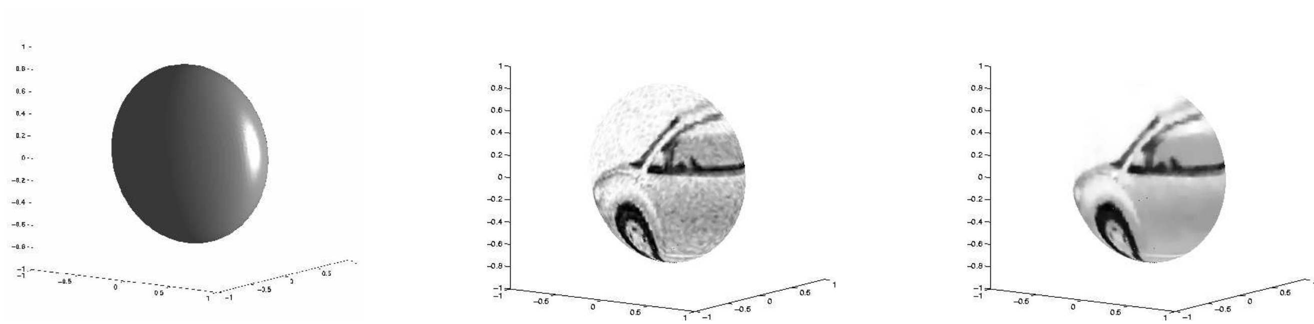
Somewhat extreme example

A 2D surface in 5D, represented by

$$\{(x_1,\cdots,x_5):\phi_1(x_1,\cdots,x_5)=0,\phi_2(x_1,\cdots,x_5)=0,\phi_3(x_1,\cdots,x_5)=0\}$$



Playing with quantities defined on surfaces

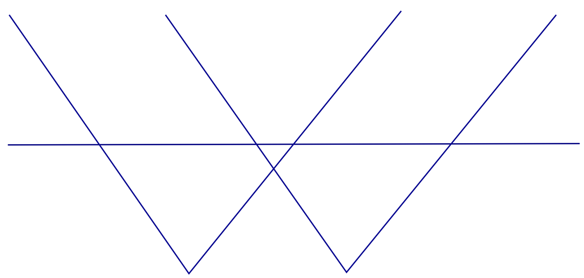


Ref: LT Cheng's Thesis, Bertalmio, Cheng, Sapiro

Set operations

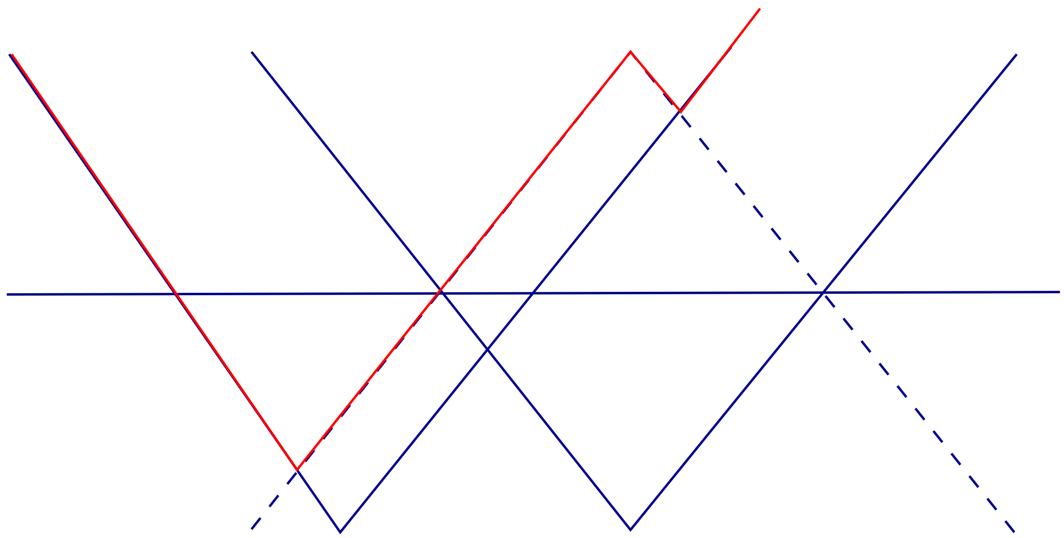
$A = \{\phi < 0\}, B = \{\psi < 0\},$

- union: $A \bigcup B = \{\phi < 0 \ or \ \psi < 0\} = \{\min(\phi, \psi) < 0\}$
- intersection: $A \bigcap B = \{\phi < 0 \text{ and } \psi < 0\} = \{\max(\phi, \psi) < 0\}$
- subtractions: $A \setminus B = \{\phi < 0 \text{ and } \psi > 0\}$=$\{\max(\phi, -\psi) < 0\}$
- complements: $A^c = \{\phi \geq 0\} = \{-\phi \leq 0\}.$

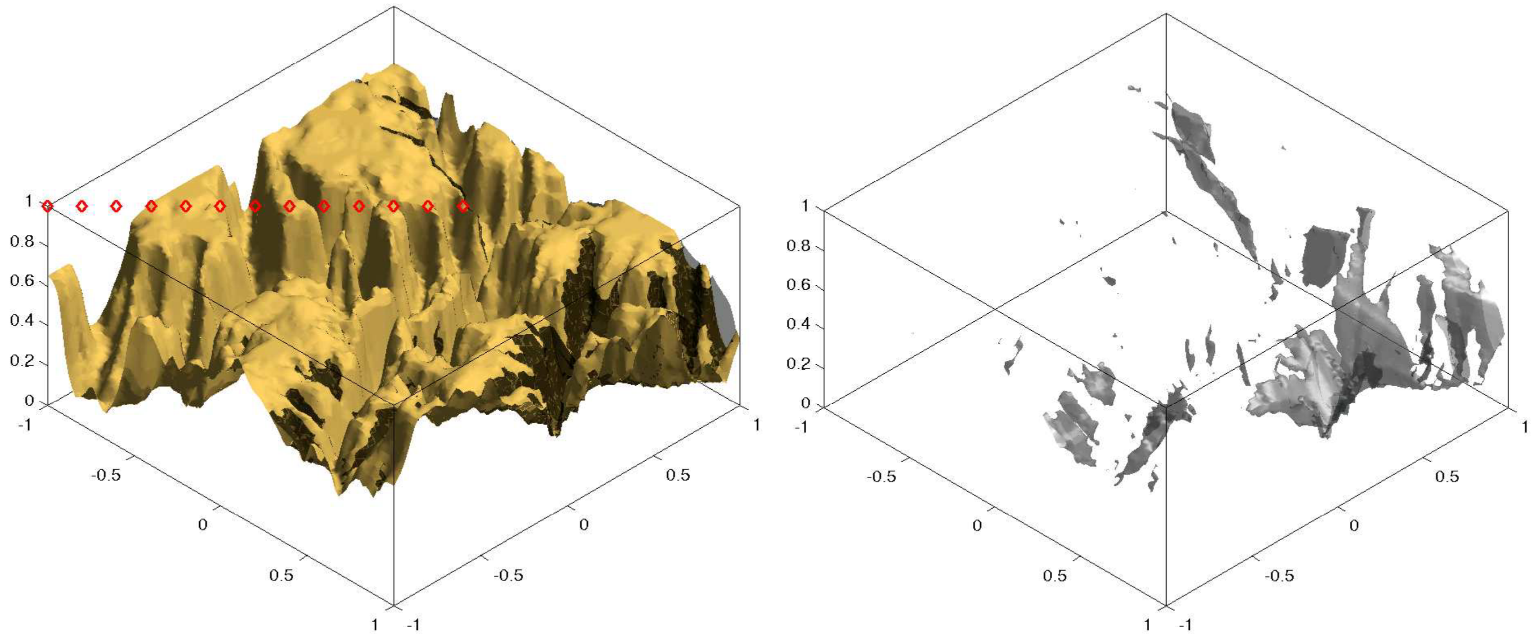


A closer look

Subtractions: $A \setminus B = \{\phi < 0 \text{ and } \psi > 0\}$=$\{\max(\phi, -\psi) < 0\}$



More example

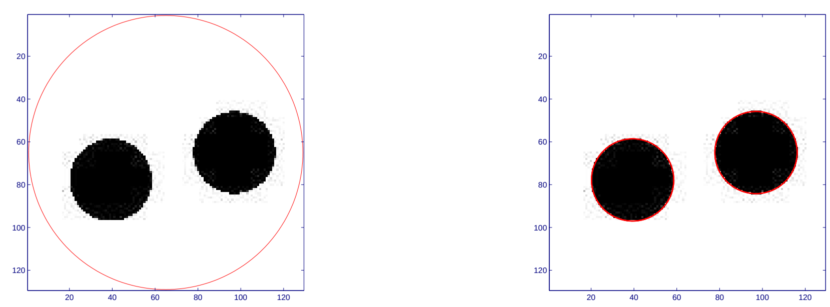


How do I see the surfaces

- MATLAB: contour, isosurface
- Other tools: VTK, IBM OpenDx, ray tracer
- NPR: non-photo-realistic rendering, see e.g. Zorin

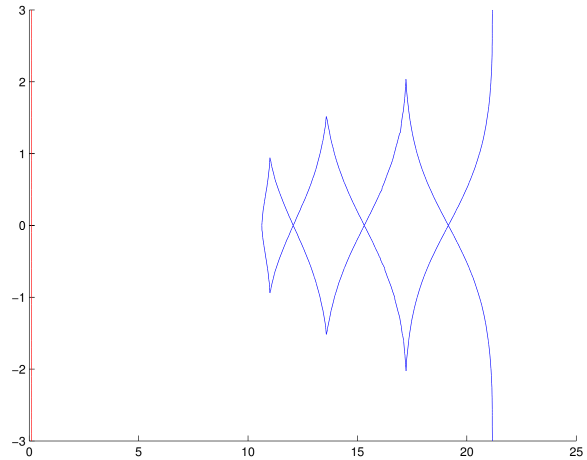
Dynamics — tracking the motion of an interfaces

- Motions governed by PDEs, mostly of Hamilton-Jacobi type
  - derivatives have jumps, need to handle with care
- Topological changes

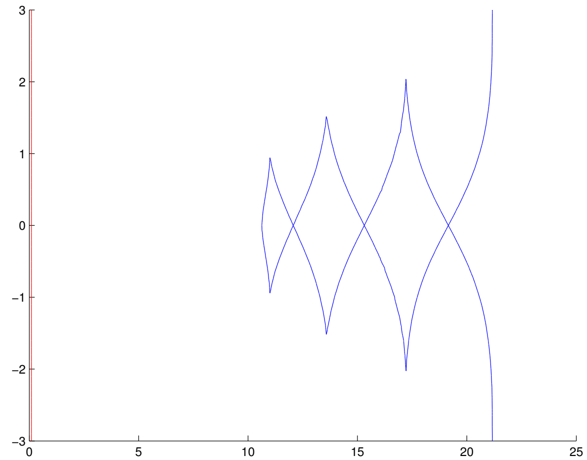


Can also prevent topological changes. (T. Cecil)

- Self-interpolating properties of the PDE approach

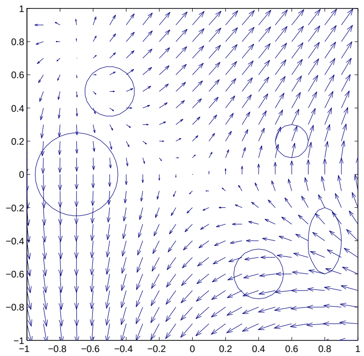


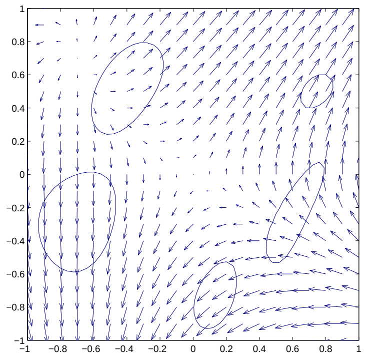
- Self-interpolating properties of the PDE approach

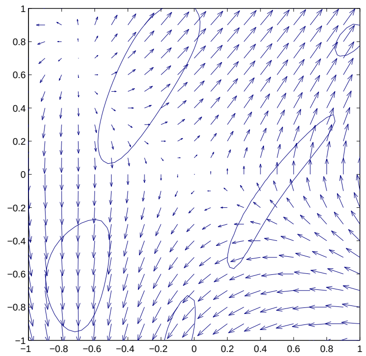


- General algorithm
- Discretizations
- Equations
- Theory

Example


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1 −0.8 −0.6 −0.4 −0.2 0 0.2 0.4 0.6 0.8 1


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1 −0.8 −0.6 −0.4 −0.2 0 0.2 0.4 0.6 0.8 1


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
−1 −0.8 −0.6 −0.4 −0.2 0 0.2 0.4 0.6 0.8 1

General Level Set algorithm

See the back of your shampoo bottle: apply shampoo/conditioner, lather, rinse and repeat

General Level Set algorithm

See the back of your shampoo bottle: apply shampoo/conditioner,
lather, rinse and repeat

---

1. (Re)Initialize ϕ (if necessary)

2. Extend v_n to the whole computational domain (if necessary)

3. Discretize and evolve

$$\phi_t + v_n|\nabla\phi| = 0.$$

4. Repeat

The Main Equation
(How to Move a Curve implicitly?)

$$\gamma(t) = (x(t), y(t)),\ \ \phi(\gamma(t), t) = 0 \text{ for all } t.$$

$$\Longrightarrow \frac{d}{dt}\phi = \underbrace{\phi_t(\gamma, t) + \gamma'(t) \cdot \nabla_{x,y}\phi(\gamma, t)}_{\text{only valid on } \gamma(t)!!} = 0$$

The Main Equation
(How to Move a Curve implicitly?)

$$\gamma(t) = (x(t), y(t)),\ \phi(\gamma(t), t) = 0 \text{ for all } t.$$

$$\Longrightarrow \frac{d}{dt}\phi = \underbrace{\phi_t(\gamma, t) + \gamma'(t) \cdot \nabla_{x,y}\phi(\gamma, t)}_{\text{only valid on } \gamma(t)!!} = 0$$

$$\Longrightarrow \phi_t(x, y, t) + v(x, y, t) \cdot \nabla_{x,y}\phi(x, y, t) = 0.$$

v need to be defined on the whole domain!

Key Equations

- $\phi_t + v \cdot \nabla\phi = 0.$
- $\phi_t + v_n|\nabla\phi| = 0, \;\; v = v_n\vec{n} + v_\tau\vec{\tau}$

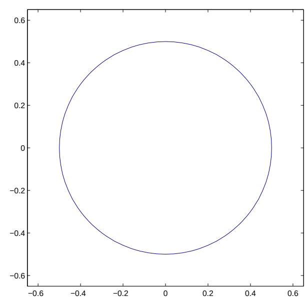
Key Equations

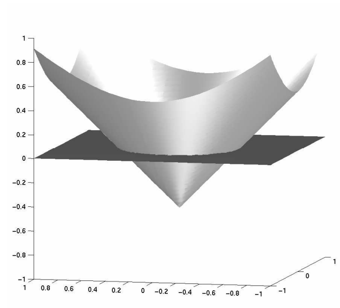
- $\phi_t + v \cdot \nabla\phi = 0.$
- $\phi_t + v_n|\nabla\phi| = 0, \;\; v = v_n\vec{n} + v_\tau\vec{\tau}$

In general,

$$\phi_t = -H(x, y, t, \nabla\phi)$$

H determines how the value of ϕ should change in time.





Homogeneity

Again, the general equation:

$$\phi_t + H(x, \nabla\phi) = 0.$$

A nice property to have:

$$H(x, \lambda\nabla\phi) = \pm\lambda^p H(x, \nabla\phi).$$

In particular, $p = 1$. This translates into: motion is invariant under scaling of the level set functions.

Eg. TV denoising and mean curvature motion:

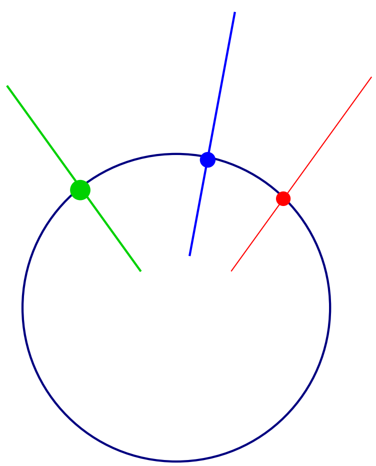
$$\phi_t - \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|}|\nabla\phi| = 0.$$

Velocity extension

Need to define v_n throughout the computational domain.

$$\nabla w \cdot \nabla \phi = 0 \quad BC: \; w \text{ given on } \Gamma.$$

Quantities do not change in the direction of the gradient $\nabla\phi$; w is constant along the characteristics.



This can be solved by time iterations or the “generalized closest point method” (R. Tsai, JCP 178, 2001)

Building/Re-shaping the level set function (1)

- From explicit surfaces (triangulation etc) to distance functions: e.g. Tsai JCP 178, 2002
- Given ϕ_0, want to keep its zero level set unchanged.
  - Solve the Eikonal equation:

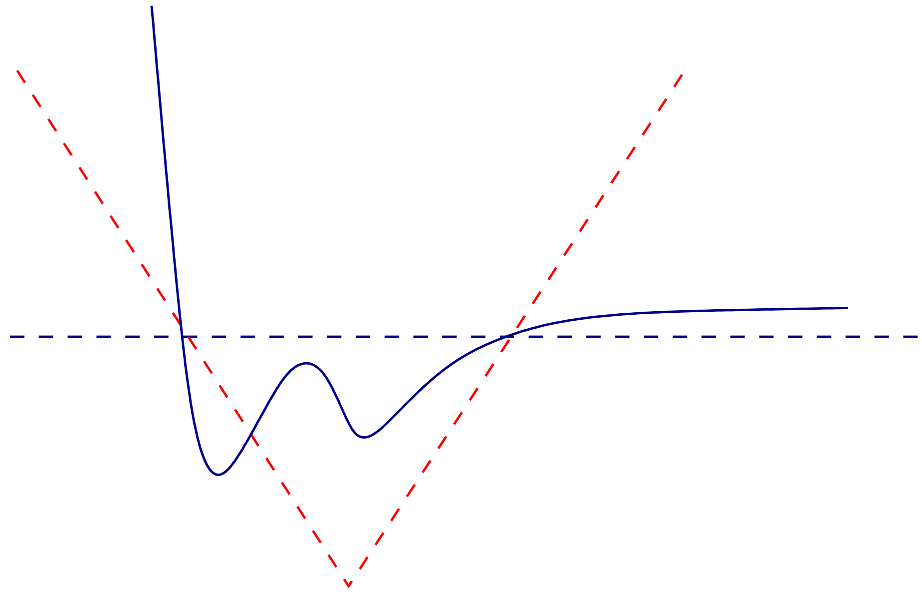
$$|\nabla\phi| = 1, \text{ BC: } \phi = 0 \text{ wherever } \phi_0 = 0,$$

    See e.g. The fast sweeping methods: Tsai et al (SINUM 2002), Kao et al. Fast marching method (Tsitsiklis 1995, Sethian 1996, Sethian-Vladimirsky 2002)
  - Solve the distance reinitialization equation:

$$\phi_t + \mathrm{sgn}(\phi_0)(|\nabla\phi| - 1) = 0, \;\; \phi(x, t = 0) = \phi_0(x).$$

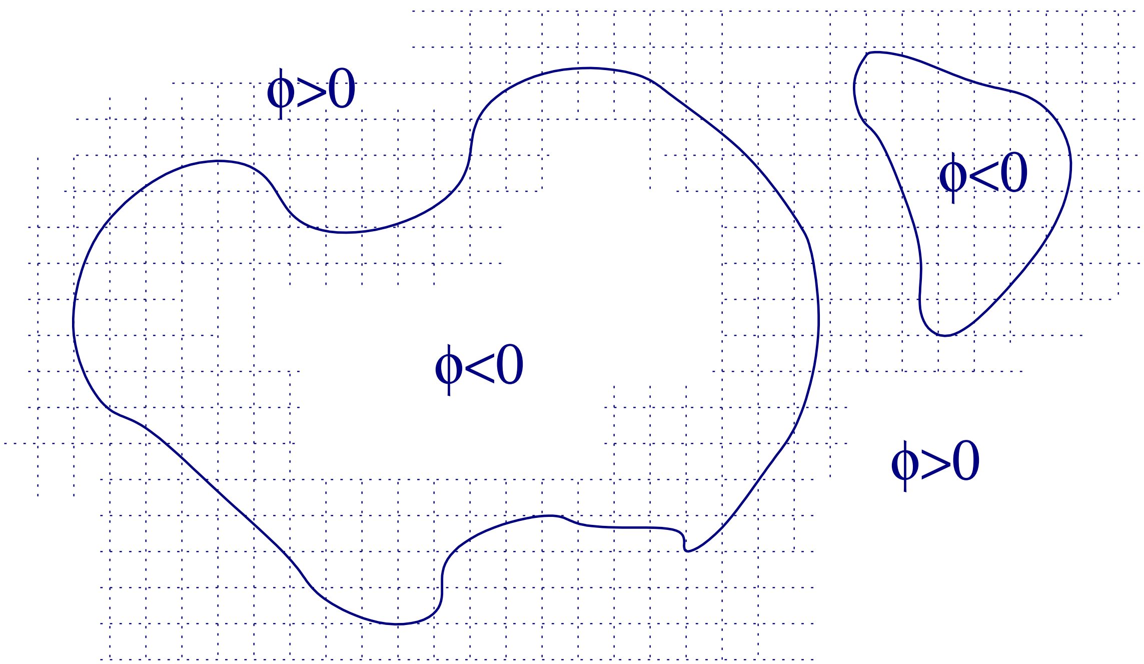
Building/Re-shaping the level set function (2)

Large and small gradients in ϕ create numerical instability and resolution issues.



$$\phi_t + \mathrm{sgn}(\phi_0)(|\nabla\phi| - 1) = 0, \;\; \phi(x, t = 0) = \phi_0(x).$$

What really happens computationally...



$\phi_t = -H(x,y,t,\nabla\phi)$ solved numerically on the grid.

Discretization (1)

1. Approximation of derivatives:

$$u_x: \quad p_- = D_-^x u_i = \frac{u_i - u_{i-1}}{\Delta x}, \;\; p_+ = D_+^x u_i = \frac{u_{i+1} - u_i}{\Delta x},$$

$$u_{xx}: \quad D_+^x D_-^x u_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}.$$

Higher order ENO type approximation: $[p_-, p_+] = \mathrm{weno}(\phi, i, j)$

Discretization (1)

1. Approximation of derivatives:

$$u_x: \quad p_- = D_-^x u_i = \frac{u_i - u_{i-1}}{\Delta x}, \; p_+ = D_+^x u_i = \frac{u_{i+1} - u_i}{\Delta x},$$

$$u_{xx}: \quad D_+^x D_-^x u_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}.$$

   Higher order ENO type approximation: $[p_-, p_+] = \mathrm{weno}(\phi, i, j)$

2. Approximation of the Hamiltonian $H(\phi_x)$ by different fluxes:

$$H(\phi_x) \sim \hat{H}(p_-, p_+).$$

Discretization (2)

- Upwinding: the Godunov Hamiltonian H^G for $\phi_t + v_n(x)\sqrt{\phi_x^2}$ is

$$H^G(p_-, p_+) = v_n \cdot \begin{cases} \sqrt{\max(\max(p_-, 0)^2, \min(p_+, 0)^2)} & v_n \geq 0 \\ \sqrt{\max(\min(p_-, 0)^2, \max(p_+, 0)^2)} & \text{otherwise} \end{cases}.$$

The form of H^G changes according to the H at hand. Bardi-Osher, SIAM J. Math. Anal. 1991.

Discretization (2)

- Upwinding: the Godunov Hamiltonian H^G for $\phi_t + v_n(x)\sqrt{\phi_x^2}$ is

$$H^G(p_-, p_+) = v_n \cdot \begin{cases} \sqrt{\max(\max(p_-,0)^2, \min(p_+,0)^2)} & v_n \geq 0 \\ \sqrt{\max(\min(p_-,0)^2, \max(p_+,0)^2)} & \text{otherwise} \end{cases}.$$

The form of H^G changes according to the H at hand. Bardi-Osher, SIAM J. Math. Anal. 1991.

- Local Lax-Friedrichs H^{LLF}:

$$H^{LLF}(p_-, p_+) = H(\frac{p_+ + p_-}{2}) - \frac{1}{2}\alpha^x(p_+, p_-)(p_+ - p_-),$$

where $\alpha^x((p_+, p_-) = \max_{p \in I(p_-, p_+)} |H_{\phi_x}(p)|$. More diffusive than H^G, but easier to evaluate.

Discretization (3)

General reference: Osher-Sethian, Osher-Shu, SINUM 1991 , Tadmor et al.

Monotone, consistent schemes $\Longrightarrow$ Convergent (viscosity solution) (Crandall-Lions, 1984)

Discretization (3)

General reference: Osher-Sethian, Osher-Shu, SINUM 1991 , Tadmor et al.

Monotone, consistent schemes $\Longrightarrow$ Convergent (viscosity solution) (Crandall-Lions, 1984)

Monotone means: for $\hat{H}(p_-, p_+)$

$$\frac{\partial H}{\partial p_\pm} \geq 0.$$

General Level Set algorithm revisited

1. (Re)Initialize ϕ (if ϕ is too steep or to flat)

$$|\nabla\phi| = 1 \text{ with suitable BC, OR}$$

$$\phi_t + \mathrm{sgn}(\phi_0)(|\nabla\phi| - 1) = 0.$$

2. Evaluate and extend v_n to the whole computational domain (if necessary)

3. Discretize (5th order WENO with Godunov or LLF flux) and evolve (TVD Runge-Kutta)

$$\phi_t + v_n|\nabla\phi| = 0.$$

4. Goto 1

General Level Set algorithm revisited

1. (Re)Initialize ϕ (if ϕ is too steep or to flat)

$$|\nabla\phi| = 1 \text{ with suitable BC, OR}$$

$$\phi_t + \text{sgn}(\phi_0)(|\nabla\phi| - 1) = 0.$$

2. Evaluate and extend v_n to the whole computational domain (if necessary)

3. Discretize (5th order WENO with Godunov or LLF flux) and evolve (TVD Runge-Kutta)

$$\phi_t + v_n|\nabla\phi| = 0.$$

4. Goto 1

All computations can be done only in a thin tube $\{|\phi| \leq \varepsilon\}$. Local level set method (Peng et al. JCP 155, 1999)

Solution theory

Viscosity solution theory: Crandall-Lions, Ishii, Evans, Souganidis, Giga

References:

- The original paper of Crandall-Lions (Trans. Amer. Math. Soc. 1983)
- Users guide: Crandall et al (Bull. Amer. Math. Soc. V27 (1), 1992)
- G. Barles, Springer-Verlag 1993
- Bardi, Capuzzo-Dolcetta 1997
- Bardi,Crandall, Evans, Soner and Souganidis, 1997
- Y. Giga's new book

Caveat

$$u_t + H(t, x, u, Du, D^2u) = 0$$

- Viscosity solution works only if H_u is of one sign! Otherwise, discontinuities may occur.

Caveat

$$u_t + H(t, x, u, Du, D^2u) = 0$$

- Viscosity solution works only if H_u is of one sign! Otherwise, discontinuities may occur.
- H or its arguments, e.g. $\frac{p}{|p|}$, may be to be regularized for computation.

Caveat

$$u_t + H(t, x, u, Du, D^2u) = 0$$

- Viscosity solution works only if H_u is of one sign! Otherwise, discontinuities may occur.

- H or its arguments, e.g. $\frac{p}{|p|}$, may be to be regularized for computation.

  E.g. Wulff flow: $\phi_t + H(\nabla\phi/|\nabla\phi|) = 0$.

$$\vec{n} \sim \vec{n}_\varepsilon \frac{p}{\sqrt{p^2+\varepsilon^2}}, \text{ or } \tanh(\varepsilon^{-1}p)$$

Which is better?

Ref: Tornberg-Engquist UCLA Cam Report, 2002

Related computational techniques

- WENO (Jiang-Peng, SIAM J. Sci. Comput., 2000)
- Central Scheme for HJ (Lin-Tadmor, SIAM J. Sci. Comput, 2000)
- Osher-Shu SIAM J. Numer. Anal. 1993
- Generalized closest point methods (Tsai, JCP 2002)
- Fast Sweeping Methods (Tsai et al SINUM 2002, Kao et al.)
- Fast marching method (Tsitsiklis 1995, Sethian 1996, Sethian-Vladimirsky 2002)

Fields of applications

- Image sciences
- Computer graphics and computer vision
- Materials sciences
- Optimal control
- Geometrical optics
- Theoretical sides
- Inverse problems
- And many more ...

A simple movie

Challenges

- High dimension computation
- Multiphase calculation
- Solution theory for general Hamilton-Jacobi equations
  (Ref. Ishii, Giga, Tsai-Osher-Giga, Math Comp. 2002)
- Solution theory for system of Hamilton-Jacobi equations
  (Ref. Burchard, Color TV image restoration, UCLA CAM Report)
- Theory for higher order equations